NOCSTAT Command

For programs optimized with the Natural Optimizer Compiler, certain statements can be directly converted into machine code when cataloged. As a result, when executing the optimized objects with Natural at runtime, the performance can be improved considerably.

The NOCSTAT command analyses cataloged programming objects and provides statistical information to help decide whether program statements benefit from optimization with the NOC and, if so, to what extent they can be optimized.

If a program is cataloged (STOW, CATALL), the Natural compiler generates an internal (pseudo) object code based on the statements in the source program. In most cases, one source statement is transformed into one pseudo-code instruction. However, for complex statements, such as FOR and REPEAT, several pseudo-code instructions are generated. The NOCSTAT analyses are based on the generated pseudo-code instructions. Therefore, the number of statements indicated in the statistical reports may exceed the number of statements in the source program.

This section covers the following topics:

- Invoking NOCSTAT
- Generating Reports
- Report Formats
- Batch Execution

Invoking NOCSTAT

▶ To invoke the NOCSTAT command

- Enter the direct command NOCSTAT.
  The main NOCSTAT screen is displayed:

```
 14:02:01              ***** NATURAL NOCSTAT COMMAND *****           2000-09-04


  Name ................. _____
  Library .............. SAGTEST_


  NOCable Objects only .. _

  Output Report ........ X Statement Category
                         _ Statement Type
                         _ Code Profile

  Output Destination .... X Screen
                          _ CSV to Work File 1
                          _ XML to Work File 1
                           with XSL _____

  Progress Control ...... X


  Command ===>
  Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10--PF11--PF12---
        Help      Exit                                                     Canc
```

To obtain field-specific help information, either enter a question mark in the relevant field and press ENTER, or place the cursor in the field and press PF1. Press PF3 to exit NOCSTAT.

Generating Reports

You can generate statistical reports for a single program or a set of programs. If you analyze more than one program at a time, the reports are produced in series. When you have finished looking at one report, press ENTER to view the next report.

The main NOCSTAT menu provides the following options:

Field	Explanation
Name	Enter a name or a range of names to specify the program(s) you want to examine: <table><tr><td></td><td>*value* is any combination of one or more characters.</td></tr><tr><td>*value*</td><td>Single program.</td></tr><tr><td>*</td><td>All programs.</td></tr><tr><td>*value**</td><td>All programs whose names begin with *value*.</td></tr><tr><td>*value>*</td><td>All programs whose names are greater/equal *value*.</td></tr><tr><td>*value<*</td><td>All programs whose names are less/equal *value*.</td></tr></table>
Library	Enter the name of a library or specify a range; the same applies as described for the Name field above. The current library is the default.
NOCable Objects only	Mark this option to exclude programs already compiled with the NOC. Otherwise, the NOCSTAT command selects all Natural programs specified in the Name and Library fields by default, including NOC-compiled programs.
Output Report	Mark any of the options to select statements by category, type or code profile. See Statement Category, Statement Type and Code Profile below.

Field	Explanation
Output Destination	Mark any of the following options to determine the output format and destination:
	Screen — Online display.
	CSV to Work File 1 — Generates spreadsheets with comma-separated values. Use the file extension ".csv" to write the work file directly to your PC for further processing.
	You can only route reports to a PC if Entire Connection is installed.
	XML to Work File 1 — Generates XML documents. Use the file extension ".xml" to write the work file directly to your PC for further processing.
	If a value is entered in the field "with XSL", a processing instruction is added at the top of the XML output document:
	`<?xml-stylesheet type="text/xsl" href="value"?>`
	The *value* entered should be the absolute or relative URL of the style sheet, for example, "nocstat.xsl" or "http://natural.software-ag.de/nocstat.xsl".
	The processing instruction causes the document to be transformed according to the given style sheet when it is viewed by an XSLT-capable browser or transformed by a batch XSLT run. A typical use of this feature is to convert the output XML to an HTML page.
	There are two XSLT style sheets delivered with Natural as text members NOCSTLS1 and NOCSTLS2 in the Natural library SYSEXUEX on the FNAT system file.
	NOCSTLS1 provides formatting instructions for report type Statement Category, NOCSTLS2 for report type Statement Type as described below.
	Download the style sheets with file extension ".xsl" to the same directory in which the XML work files are stored.
	You can only route reports to a PC if Entire Connection is installed.
	See also Progress Control and Batch Execution below.
Progress Control	Only applies to Work File 1 output destinations. If this option is marked, a brief message appears online for each program listed in the report generated.

Report Formats

You can choose between three output formats described below to display the statistics NOCSTAT provides for the statements analyzed. Different report layouts are produced for programs already optimized with the NOC and for programs to be considered for optimization. The example reports below show the difference. Press PF3 to interrupt report processing and return to the NOCSTAT menu.

Below is information on:

- Statement Category
- Statement Type
- Code Profile

Statement Category

The statistical report generated with the option "Statement Category" lists various categories of statements with the corresponding number of occurrences and the total number of statements already optimized or suitable for optimization, depending on whether or not the program was optimized with the NOC.

Example of NOC-Optimized Program:

```
14:07:17              ***** NATURAL NOCSTAT COMMAND *****              2000-09-04
                 Library SAGTEST  Name NOCTEST1 Type Program


     MCG Options: (ON,OVFLW,INDX,MIX,IO)


       Database Loop:        0
     Database Simple:        0
     SORT / WORK I/O:       29
        FOR / REPEAT:        0
     Screen / Printer:      59
   String Manipulation:      6
      Arith / Logical:       0
        Program Calls:       3
     Control Transfer:      49
          Block Start:      25
      Set Environment:       2
     System Functions:       0
        Miscellaneous:       0


     Total Statements:     949
        NOC optimized:     762  ( Ratio:  80 % )
      Longest NOC Run:     180  Statements
```

Example of Program without NOC Optimization:

```
 14:13:01                ***** NATURAL NOCSTAT COMMAND *****            2000-09-04
                    Library SAGTEST  Name NOCTEST2 Type Program

                          No NOC      NOCable
                         --------     --------
             Database Loop:      0            0
           Database Simple:      0            0
           SORT / WORK I/O:      0            0
              FOR / REPEAT:      0            5
           Screen / Printer:    57            0
        String Manipulation:     4            8
            Arith / Logical:      0          491
              Program Calls:      3            0
           Control Transfer:     19           69
                Block Start:     15            0
            Set Environment:      0            0
            System Functions:     0            0
              Miscellaneous:      0            0

                     Total Statements:      672
                       NOC optimizable:     573  ( Ratio:  85 % )
                       Longest NOC Run:     192  Statements
```

Report Columns and Fields:

Column	Explanation
No NOC	Statements not suitable for optimization.
NOCable	Statements suitable for optimization.
Field	
Database Loop	The number of database statements that generate a processing loop, such as FIND and READ.
Database Simple	Database statements that do not generate a processing loop, such as STORE, UPDATE, DELETE and GET.
SORT / WORK I/O	SORT and WORK file statements.
FOR / REPEAT	Statements generating loops.
Screen / Printer	Screen and printer I/O, such as WRITE, DISPLAY and INPUT.
String Manipulation	String statements, such as EXAMINE and COMPRESS.
Arith / Logical	Arithmetic and logical statements, such as MOVE, COMPUTE and IF.
Program Calls	Transfer of control to a subroutine or subprogram, such as PERFORM, CALLNAT and FETCH.
Control Transfer	Jumps within the program, such as ESCAPE BOTTOM, FOR and REPEAT loops.
Block Start	Non-executed statements that demarcate code blocks, such as DEFINE SUBROUTINE and AT END. These statements are never optimized because they are never executed.
Set Environment	Statements that set the environment, such as SET CONTROL, SET GLOBALS and SET KEY.
System Functions	Statements, such as TOTAL, SUM, COUNT, MAX, MIN and *COUNT.
Miscellaneous	Pseudo-code statements not relevant for optimization and, therefore, ignored by the NOC.
Totals	
Total Statements	The total number of statements found in the program. This number may not correspond to the actual source statements as described in the introduction to NOCSTAT Command above.
NOC optimized	For an optimized program, these are the actual pseudo-code statements (as described in the introduction to NOCSTAT Command above) that have been NOC-optimized to machine code.
NOC optimizable	For non-optimized programs, this is the possible number of statements that could be optimized. The figure may be slightly higher than the actual number, since certain factors are not considered in the NOCSTAT program. For example, a SUBSTRING statement that has more than four arrays will be indicated as "optimizable" though it will not be optimized.
Ratio	Relation between Total Statements and NOC-optimized statements or Total Statements and NOC-optimizable statements in percent.
Longest NOC Run	NOC-optimized program: The number of contiguous optimized statements - the fewer fragment sequences, the better the performance.
	Non-optimized program: The number of contiguous statements to be expected if the program were optimized.

Statement Type

The statistical report generated with the option "Statement Type" lists single statements with the corresponding number of occurrences and the NOC coding generated for optimized objects.

Example of NOC-Optimized Program:

```
09:21:45             ***** NATURAL NOCSTAT COMMAND *****         2000-09-06
                Library SAGTEST  Name NOCTEST1 Type Program


        Statement             Number
------------------------    --------
 DB AT CONDITION                  6
 READ/WRITE WORK FILE            29
 EXAMINE                          6
 WRITE                           51
 INPUT                            3
 NEWPAGE                          2
 REINPUT                          3
 FIND                            1
 READ                             2
 NOC CODE                       760
 BLOCK START                     18
 ON ERROR                         1
 END                             1
 STOP                             2
 RETURN                          3
 RETURN INLINE                   15
 ESCAPE ROUTINE                   3
 ESCAPE ROUTINE IMMEDIATE         1
 MORE
```

Example of Program without NOC Optimization:

```
09:23:15             ***** NATURAL NOCSTAT COMMAND *****         2000-09-06
                Library SAGTEST  Name NOCTEST2 Type Program


        Statement            No NOC   NOCable
------------------------    --------  --------
 DB AT CONDITION                  6         0
 MOVE/COMPUTE/ASSIGN              0       371
 EXAMINE                          4         0
 COMPRESS                         0         7
 WRITE                           47         0
 INPUT                            2         0
 NEWPAGE                          2         0
 REINPUT                          6         0
 FIND                             1         0
 READ                             1         0
 HISTOGRAM                        1         0
 ELSE/CLOSE LOOP                  0        55
 LOOPEND FOR/REPEAT               0         5
 BLOCK START                      8         0
 ON ERROR                         1         0
 END                             1         0
 STOP                             2         0
 RETURN                           2         0
 MORE
```

Code Profile

The statistical report generated with the option "Code Profile" displays contiguous sequences of statements grouped by categories in a source program suitable for optimization, or lists the NOC coding generated for an optimized program. Occurrences are highlighted.

Example of NOC-Optimized Program:

```
09:59:04                ***** NATURAL NOCSTAT COMMAND *****          2000-09-06
                  Library SAGTEST  Name NOCTEST1 Type Program


  Line    Statement
  -----   ------------------------
  0000    ON ERROR
  0000    MCG OPTIONS
  0045    MCG OPTIONS
  0050    NOC CODE
  0050    NOC CODE
  0050    NOC CODE
  0050    NOC CODE
  1110    SET KEY
  1140    NOC CODE
  1140    NOC CODE
  1145    NOC CODE
  1145    NOC CODE
  1150    NOC CODE
  1150    NOC CODE
  1155    NOC CODE
  1155    NOC CODE
  1160    NOC CODE
  1160    NOC CODE
  MORE
```

Example of Program without NOC Optimization:

```
10:01:36                ***** NATURAL NOCSTAT COMMAND *****          2000-09-06
                  Library SAGTEST  Name NOCTEST2 Type Program


  Line    Statement
  -----   ------------------------
  0000    ON ERROR
  0000    MCG OPTIONS
  0100    MOVE/COMPUTE/ASSIGN       <-- NOCable
  0100    MOVE/COMPUTE/ASSIGN       <-- NOCable
  0100    MOVE/COMPUTE/ASSIGN       <-- NOCable
  1920    MOVE/COMPUTE/ASSIGN       <-- NOCable
  1920    FOR                       <-- NOCable
  1920    MOVE/COMPUTE/ASSIGN       <-- NOCable
  1920    FOR/REPEAT IF             <-- NOCable
  1930    COMPRESS                  <-- NOCable
  1940    LOOPEND FOR/REPEAT        <-- NOCable
  1960    MOVE/COMPUTE/ASSIGN       <-- NOCable
  1960    MOVE/COMPUTE/ASSIGN       <-- NOCable
  1970    MOVE/COMPUTE/ASSIGN       <-- NOCable
  1970    MOVE/COMPUTE/ASSIGN       <-- NOCable
  1980    MOVE/COMPUTE/ASSIGN       <-- NOCable
  1980    MOVE/COMPUTE/ASSIGN       <-- NOCable
  1990    MOVE/COMPUTE/ASSIGN       <-- NOCable
  MORE
```

Batch Execution

Below are job examples for processing NOCSTAT reports in batch mode. After job execution, the work files generated can be transferred from host to PC for further processing with standard transfer tools.

Example Job OS/390:

```
//NOCBATCH JOB  (NOC,,,30),CLASS=K,MSGCLASS=X                        00000100
//NATEX EXEC PGM=NATBAT31,REGION=6200K,PARM=('IM=D')                 00000200
//STEPLIB  DD DISP=SHR,DSN=TESTNAT.LOAD                              00000300
//CMPRINT  DD SYSOUT=X                                               00000400
//CMWKF01  DD DSN='NOC.NOCSTAT.OUT',DISP=(NEW,CATLG),                00000500
             SPACE=(CYL,(1,1)),UNIT=SYSDA,VOL=SER=SAG001             00000600
//SYSOUT   DD SYSOUT=X                                               00000700
//CMSYNIN  DD *                                                      00000800
NOCSTAT                                                              00000900
*,library,X,,,X                                                      00001000
.                                                                    00001100
FIN                                                                  00001200
/*                                                                   00001300
```

Example Job VSE/ESA:

```
* $$ JOB JNM=NOCTST,CLASS=5,DISP=D
* $$ LST CLASS=Q,DISP=D
// JOB NOCTST
// ASSGN SYS001,DISK,VOL=xxxxxx,SHR
// DLBL CMWKF01,'NOCSTAT.FILE.ONE',0
// EXTENT SYS001,xxxxxx,1,0,1,150
// EXEC NAT234BA,SIZE=NAT314BA,PARM='SYSRDR'
IM=D,OBJIN=R
/*
ADARUN DBID=185
/*
NOCSTAT
*;library;X; ; ; ;X;
.
FIN
/*
/&
```

Example Job BS2000/OSD:

```
/.BAT234 LOGON  NAT,1
/        SYSFILE SYSOUT=NAT314.OUT
/        SYSFILE SYSLST=NAT314.LST
/SKIP   .NOP000
=================================================================
      NAME   : E.NAT314        S T A R T  B A T C H  N A T U R A L
=================================================================
/.NOP000 REMARK
/        OPTION  DUMP=YES,MSG=FL
/        FILE    NOCSTAT.OUT,LINK=W01
/        FILE    ADAUSER  ,LINK=DDCARD
/        FILE    $SAG.ADA623.MOD      ,LINK=BLSLIB00
/        SYSFILE TASKLIB=MOD234
/        SYSFILE SYSDTA=(SYSCMD)
/        FILE    NAT314.CMPRMIN,LINK=CMPRMIN
/        DCLJV   NATJV1,LINK=*NATB2JV
/        FILE    $NAT.ADALNK.PARMS,LINK=DDLNKPAR
/        REMARK  %%%%%%%%%  BATCH-PHASE   %%%%%%%%%%%%%
/        EXEC    NAT314
NOCSTAT
*,ADE,X, , , ,X, , ,X
.
FIN
```